

Juin 2000

Modélisation et maillage d'un parapente en vol



Bertrand DENEL
Patrick FOURCADE
Sous la direction de **P. PUISEUX**

SOMMAIRE

I. Introduction : qu'est-ce qu'un parapente ? *3*

A. Bref historique **3**

B. Schémas et vocabulaire **4**

II. Le projet global *6*

III. Les programmes Corde et Infierno *8*

A. Structure de données et concepts utilisés **8**

B. Corde **8**

1. But 8
2. La structure d'un fichier en-tête 9
3. Les lois utilisées 10
   a. Inter-nervure 10
   b. Rayon de déploiement en envergure 10
4. La structure d'un fichier .in 11
5. Syntaxe d'appel 12

C. Le programme Infierno **13**

1. But 13
2. Méthode de déploiement 13
3. Les formats de sortie et leur utilisation 15
   a. Le format CMARC 15
   b. Les formats de visualisation pour le parapente 20
      i. Le format GNUPLOT 20
      ii. Le format DXF 21
      iii. Le format PLOTMTV 21
      iv. Le format MATLAB 22
   c. Les informations complémentaires 25
      i. La sortie CELLULE 25
      ii. La sortie POINT_R 27
      iii. La sortie SUSPENTE 29
4. La syntaxe d'appel 30

IV. Utilisations possibles avec CMARC et Hippolyte *31*

V. Conclusion *35*

Annexes *36*

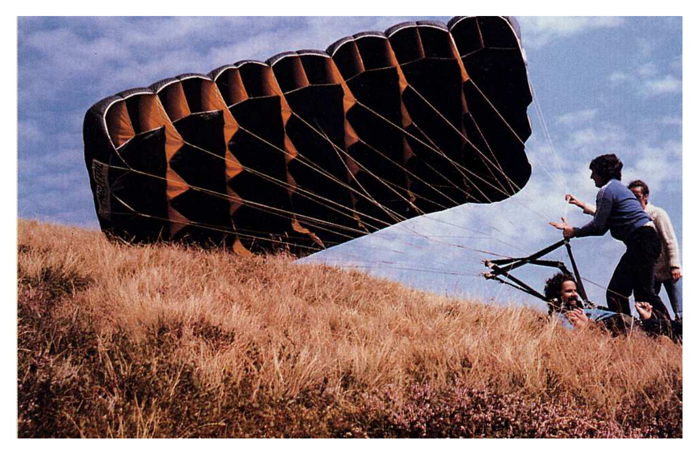
   a. Le programme de visualisation MATLAB 36
   b. Syntaxe d'appel de mtb2obj_full 37

Bibliographie et sites WEB *38*

I. INTRODUCTION : QU'EST-CE QU'UN PARAPENTE ?

A. BREF HISTORIQUE

Le parapente est né dans les années 70, dans la tête puis dans les jambes de deux parachutistes adeptes de l'atterrissage de précision, Jean-Claude Bétemps et Gérard Bosson.
A la recherche d'une méthode d'entraînement bon marché, ils sont les premiers à avoir eu l'idée d'utiliser leurs parachutes pour décoller d'une pente suffisante.



Premier décollage sur les pentes du Pertuiset

Depuis le Pertuiset à Mieussy, lieu des premières expérimentations, le parapente va rapidement envahir la planète entière et conquérir les sommets les plus prestigieux : le Mont Blanc, l'Aconcagua (plus haut sommet du continent Américain , presque 7000 m), le Gasherbrum et l'Everest (sommets himalayens dépassant les 8000 m) pour ne citer qu'eux.



11 juillet 1985, Pierre Gevaux décolle du Gasherbrum II

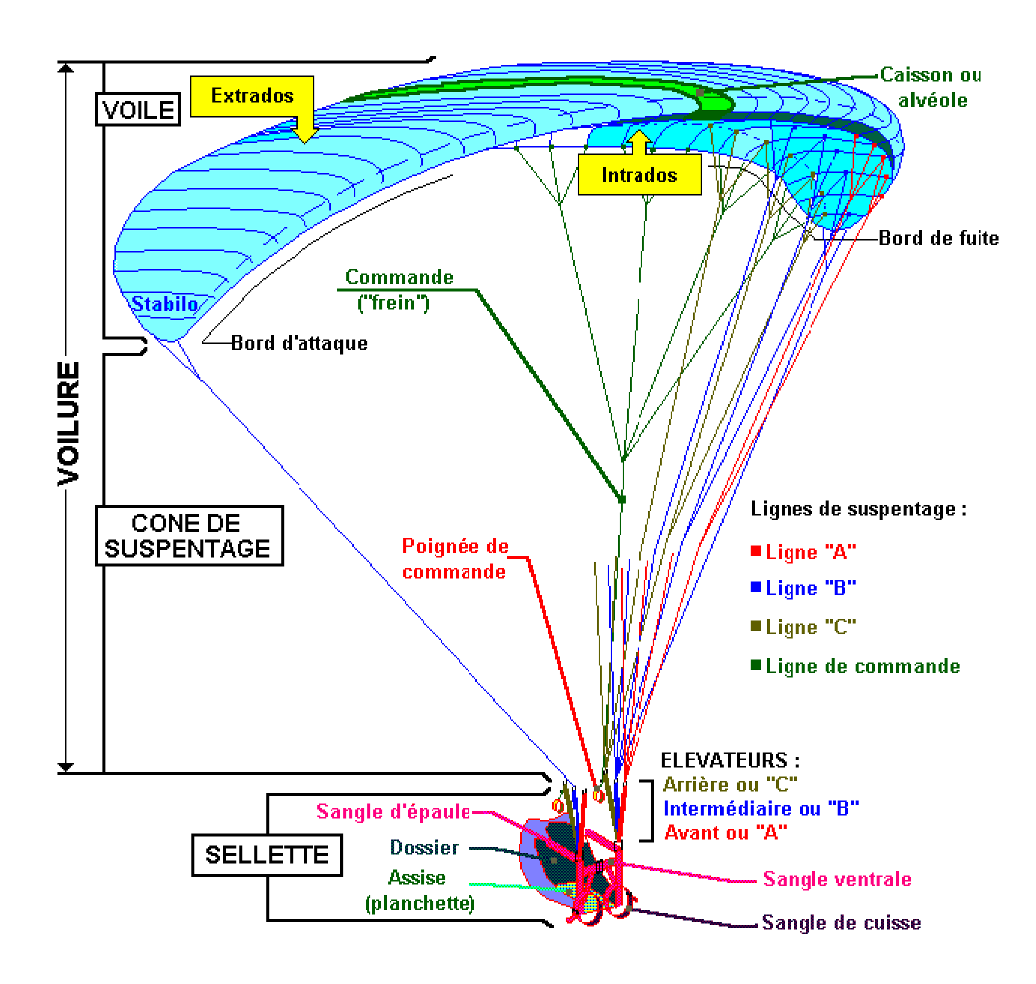
Il faut dire que le parapente est à ce jour le sport aérien le plus accessible, vu sa facilité d'apprentissage et la simplicité du matériel utilisé. Le parapente est en effet un aéronef de quelques kilogrammes qui se plie au fond d'un sac à dos et qui décolle en quelques mètres sur n'importe quel petit bout de pente.

Cependant, il ne faut pas oublier que derrière cette facilité apparente se cachent des problèmes bien plus complexes liés à l'aérologie, la météorologie et la mécanique des fluides.

Aussi il n' est pas étonnant que quelques adeptes de ce nouveau sport aient rapidement essayé de mettre le vol en équations afin d' améliorer les ailes et de prolonger la durée desvols. Les constructeurs ont alors commencé à comprendre la mécanique du vol et ont fait évoluer leurs voiles vers plus de performances.

B. SCHEMAS ET VOCABULAIRE

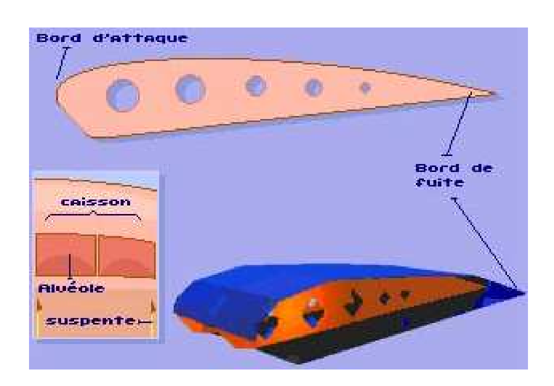
Afin de mieux comprendre ce qui suit, il paraît indispensable d'introduire dès à présent un certain nombre de notions et de termes techniques relatifs au parapente.



Comme on le voit sur le schéma, un parapente est constitué d'une *voile* et d'un *cône de suspentage*.

- **La voile**

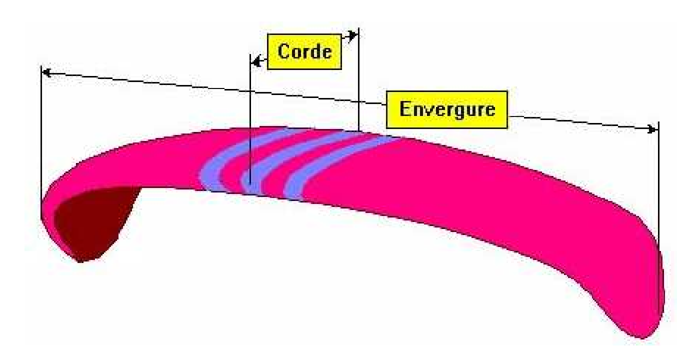
Les deux pièces principales de la voile sont *l'extrados* - la partie supérieure du profil - et *l'intrados* - la partie inférieure. Elles sont reliées entre elles par des pièces de tissu verticales qui donnent à la voile la forme de son profil : ce sont les *nervures*.
Une partie de la voile délimitée par deux nervures, l'intrados et l'extrados s'appelle un *caisson* ou *cellule*.



L'avant de la voile se nomme le *bord d'attaque*. L'arrière de la voile se nomme le *bord de fuite*. De chaque côté de l'aile, on trouve les *stabilisateurs* chargés de réduire la traînée.
Au bord d'attaque sont placées des *ouvertures* permettant aux caissons de se gonfler et de former le profil de la voile l'autorisant à s'envoler.

La distance entre le bord d'attaque et le bord de fuite est appelée *corde*. Elle n'est pas constante sur toutes les nervures, entre le centre et les bouts d'aile.

L'*envergure* est la distance maximale entre le centre et les bouts d'aile. Il y a en fait plusieurs façons de mesurer l'envergure : à plat, voile étalée sur le sol (on peut d'ailleurs de cette manière la mesurer soit à l'extrados, soit à l'intrados, soit à la corde du profil) ou en projeté, avec la voile gonflée normalement.



Par ailleurs, les cordes des différentes nervures ne sont pas dans un même plan. Chacune d'elle possède un degré de rotation autour du *point de référence*. L'angle de cette rotation est appelé *vrillage*.

Enfin, les nervures ne sont pas alignées entre elles : seuls les *points de référence* le sont. Ces points sont situés à un pourcentage fixe de corde nommé *insertion*. Par exemple, une insertion de 0% aligne tous les bords d'attaque alors que pour 100% ce sont les bords de fuite qui sont alignés.

- **Le cône de suspentage**

Comme son nom l'indique, il regroupe l'ensemble des *lignes de suspentage* ou *suspentes* qui relie la voile au pilote. Les suspentes sont regroupées au niveau de deux *maillons* séparés d'une distance nommée *écart*.

II. Le projet global

Aujourd' hui, la simulation numérique de phénomènes physiques a pris une place prépondérante dans la plupart des grandes entreprises des secteurs primaire et secondaire (aéronautique, pétrole, automobile, chimie, travaux publics...) où les décisions importantes sont de plus en plus subordonnées à des résultats de modélisation.

La conception de parapentes ne déroge pas à cette règle.

Ainsi, notre projet de maîtrise, proposé et encadré par P. Puiseux, s'insère dans une chaîne visant à simuler numériquement un parapente en vol.

Le programme de simulation 3D existe déjà : il s'agit d'une soufflerie numérique développée par la NASA et nommée CMARC.

De plus, une interface post-soufflerie facilitant l'interprétation des résultats obtenus a été développée sous le nom d'*Hipollyte* par Jérôme Sarthe (DESS Université de Savoie 1999).

Notre travail consiste donc à générer le maillage d'un parapente complet à partir d'une description géométrique plus ou moins détaillée.

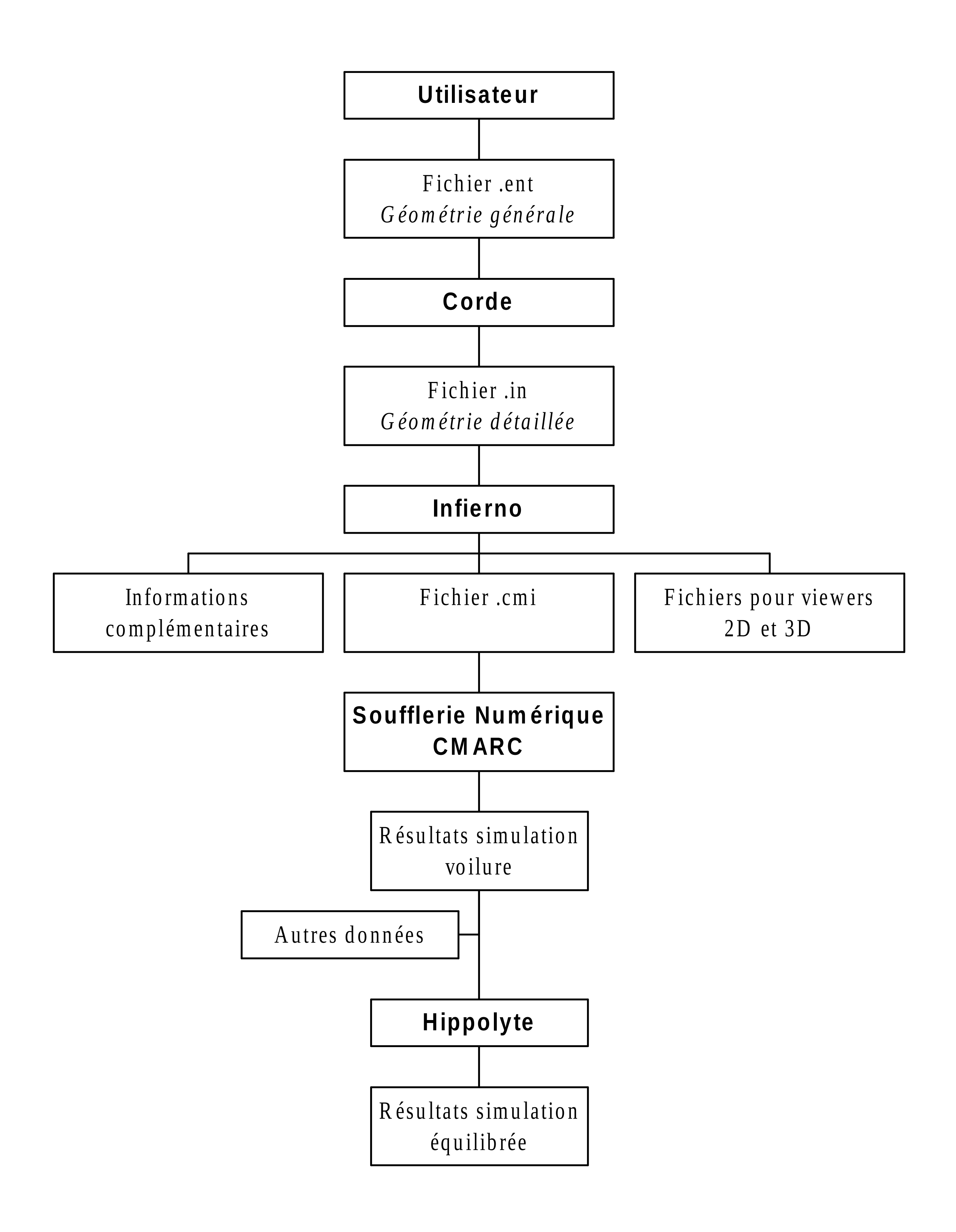
Le maillage ainsi obtenu doit répondre à deux critères :

- être utilisable par la soufflerie numérique CMARC
- permettre d'obtenir, sans trop d'efforts, une représentation graphique du parapente bâti.

Pour résoudre ce problème, nous avons développé, à partir d'un code Pascal et de quelques objets C++ existants, deux programmes :

- **Corde** qui génère une description détaillée du parapente à partir de caractéristiques géométriques générales de celui-ci.
- **Infierno** qui réalise le maillage du parapente à partir du fichier généré par Corde et qui propose des sorties au format interprétable tant par CMARC que par des « viewers » 3D.

Ces programmes ont été réalisés en C++ (sous Linux) afin de bénéficier des avantages d'un langage orienté objet. Nous pensons en particulier à la possibilité de faire facilement évoluer les objets vers d'autres plus complets et plus réalistes.



Modélisation numérique d'un parapente

III. LES PROGRAMMES CORDE ET INFIERNO

A. STRUCTURE DE DONNEES ET CONCEPTS UTILISES

La structure de données utilisée dans la première version du programme en Pascal Objet nous a semblée assez naturelle et cohérente : nous nous en sommes donc servi comme base.
Nous avons cependant été amenés à compléter ces objets (notamment la classe `Point3D`), à modifier certaines de leurs méthodes comme celle du placement des nervures qui est inédite, et à en développer de nouveaux comme la classe `Cellule` afin de modéliser plus précisément le parapente.

Ainsi, la structure que nous avons retenue est la suivante :

- On considère qu'un parapente est un ensemble de nervures et de cellules
- Une nervure, en plus de ses caractéristiques propres, est associée à un certain profil géométrique.
  Une cellule est quant à elle parfaitement définie à partir des deux nervures qui la délimitent.
- Un profil est un polygone « lissé » de façon à améliorer ses caractéristiques aérodynamiques.
- Un polygone est un ensemble de points 2D.

Chaque objet utilisé par Infierno a été développé dans le souci d'être facilement réutilisé et complété lors de développements ultérieurs.

Pour de plus amples informations sur les données, nous avons généré une documentation hypertexte détaillée de Infierno à l'aide de l'utilitaire DOXYGEN.
On y trouvera le nécessaire pour comprendre et dériver Infierno.

B. CORDE

1. But

Ce programme sert d'interface entre l'utilisateur et Infierno. En effet, il permet à l'utilisateur de créer un fichier .in à partir de quelques caractéristiques techniques du parapente souhaité.
Le fichier .in généré pourra être modifié à la main par l'utilisateur, notamment pour certains paramètres automatisés comme *décalBA_*.
Deux modes de fonctionnement sont possibles avec ce programme : le premier que nous nommerons « light » et le second « détaillé ».
En effet, le premier est destiné à un appel dit « inline » : l'ensemble des paramètres qu'entrerait l'utilisateur au clavier est contenu dans un fichier en-tête (extension ent). Quant au second mode de fonctionnement, il demande à l'utilisateur de rentrer successivement chacun de ces paramètres. A la fin de cette saisie, il est proposé de sauver la configuration dans un fichier en-tête, de façon à accélérer la prochaine saisie.

2. La structure d'un fichier en-tête

Comme énoncé ci-dessus, le fichier en-tête regroupe diverses caractéristiques du parapente. Ce paragraphe détaille donc la structure des fichiers .ent et le manuel d'utilisation de Corde.
Dans l'ordre de saisie, ces paramètres sont :

- Echelle de l'étude
- Demi-envergure (en mm)
- Corde en centre d'aile (en mm)
- Nombre de nervures par demi-aile. Un nombre maximum est fixé dans `constantes.h`. Au-delà de celui-ci, le programme stoppe.
- Coefficient de la progression géométrique inter-nervure (cf III.B.3.a page 10)
- Demi-écart (en mm)
- Suspentage (en mm) : il s'agit, en fait, de la distance entre l'origine et le point de référence de la nervure centrale. Il y a donc une légère différence avec la définition théorique.
- Coefficient de la progression géométrique du rayon de courbure (cf III.B.3.b page10)
- Insertion (en % de corde)
- Ouverture avant (en % de corde)
- Ouverture arrière (en % de corde)
- Vrillage min et max
- Nombre de rangs de suspentes par nervure
- Emplacement des points d'ancrage de ces suspentes (en % corde)

Exemple de fichier .ent :

```
#Echelle_de_l'etude
0.5
#Demi_envergure_(en_mm)
6000
#Corde_en_centre_d'aile_(en_mm)
3000
#Nombre_de_nervures_par_demi-aile
20
#Coef_prog_geometrique_distance_inter_nervure
0.99
#Demi-ecart_(en_mm)
200
#Suspentage_(en_mm)
7500
#Coef_prog_geometrique_rayon
0.97
#Insertion_(en_%_de_corde)
50
#Ouverture_avant_(en_%_de_corde)
4
#Ouverture_arriere_(en_%_de_corde)
12
#Vrillage_Min,_Max
0     4
#Nombre_de_rangs_de_suspentes
4
#Emplacement
4    20    35    70
```

3. Les lois utilisées

a. Inter-nervure

Il paraît naturel que les nervures ne soient pas réparties uniformément : les nervures en bout d'aile sont plus resserrées que les nervures centrales. C'est sur cette constatation que nous avons arbitrairement choisi des écarts inter-nervure en progression géométrique. La première nervure est placée à la verticale de l'origine et la dernière à une distance de *demi_env_* de celle-ci.
Connaissant le nombre de nervures intermédiaires, il vient la loi suivante :

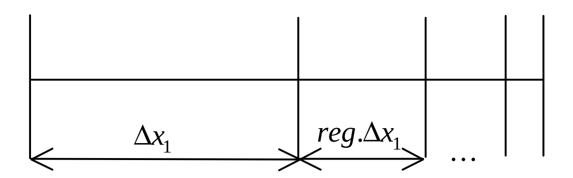
$$demi_env_ = \sum_{i=0}^{n-1} \Delta x_i = 0 + reg.\Delta x_1 + reg^2.\Delta x_1 + \ldots + reg^{n-1}.\Delta x_1 = \Delta x_1 \left(1 + reg + \ldots + reg^{n-1}\right) = \Delta x_1 \frac{1-reg^n}{1-reg}$$

On a donc :

$\Delta x_1 = demi_env_ \;.\frac{1-reg}{1-reg^n}$ où Δx_1 est la distance entre les deux premières nervures.

Les largeurs inter-nervure sont alors données par la loi :

$largeur_k = reg^{k-1}.\Delta x_1$ où $largeur_k$ est la distance entre $nervure_{k-1}$ et $nervure_k$



b. Rayon de déploiement en envergure

Pour la nervure *i*, le rayon est donné par :

$$rayon_ = (reg_2)^i . suspentage_$$

Pour une description plus détaillée de l'utilisation de ce rayon, se référer à III.C.2 page 13.

4. La structure d'un fichier .in

Le fichier .in suivant a été généré à partir du fichier .ent du paragraphe III.B.2 :

```
#Fichier_.IN_pour_programme_INFIERNO
#==================================
#Demi_envergure_(en_mm)
3000
#Corde_en_centre_d'aile_(en_mm)
1500
#Nombre_de_nervures_par_demi-aile
20
#Demi-ecart_(en_mm)
100
#Suspentage_(en_mm)
3750
#Insertion_(en_%_de_corde)
50
#Ouverture_avant_(en_%_de_corde)
4
#Ouverture_arriere_(en_%_de_corde)
12
#Vrillage_Min,_Max
0     4
#Nombre_de_rangs_de_suspentes
4
#Emplacement
4    20    35    70
#Nervure_Nø0
#Fichier,_Corde,_Largeur,_DecalBA,_Vrillage,_OuvEx.x,_OuvIn.x,_Rayon
milieu.lis    1500    0    0    0    60    180    3750
#Nombre_de_rangs_de_susp_et_valeurs
4
60    300    525    1050
#Nervure_Nø1
#Fichier,_Corde,_Largeur,_DecalBA,_Vrillage,_OuvEx.x,_OuvIn.x,_Rayon
palas.lis    1498.12    164.751    0    0.210526    59.925    179.775    3637.5
#Nombre_de_rangs_de_susp_et_valeurs
4
59.925    299.625    524.343    1048.69
#Nervure_Nø2
#Fichier,_Corde,_Largeur,_DecalBA,_Vrillage,_OuvEx.x,_OuvIn.x,_Rayon
milieu.lis    1492.48    163.103    0    0.421053    59.6992    179.098    3528.38
#Nombre_de_rangs_de_susp_et_valeurs
4
59.6992    298.496    522.368    1044.74
...
```

On remarque que l'en-tête reprend dans l'ensemble les informations du fichier .ent. Il manque cependant le facteur d'échelle qui a été appliqué aux données. Suit alors une description des paramètres propres à chaque nervure. On constate par exemple que les profils utilisés peuvent varier selon les nervures.
L'utilisateur est alors libre de modifier à la main ces paramètres.

5. Syntaxe d'appel

Lors de son exécution, Corde propose de choisir entre un mode mono-profil et un mode poly-profil. Si l'utilisateur choisit le mode mono-profil, il donne le nom du profil utilisé pour toutes les nervures. Pour l'autre mode, il doit entrer pour chaque nervure le profil correspondant.

La syntaxe générale d'appel du programme Corde est de la forme :

`Corde [-i fic_ent] -o fic_out [-p profil]`

Détail des options :

`-i` Fichier `ENTETE`
Cette option permet d'accélérer la saisie des paramètres de Corde : les caractéristiques générales du parapente sont contenues dans ce fichier.

`-o` Fichier `ENTREE`
Cette option est indispensable et indique à Corde le nom de la description détaillée du parapente à générer.

`-p` Fichier `PROFIL` utilisé
Cette option active automatiquement le mode mono-profil ; toutes les nervures auront le même profil.

En cas de problèmes, une aide est proposée par la commande : **`Corde –h`**

Par ailleurs, les extensions des fichiers sont indiquées dans le fichier `constantes.h`.

Ainsi, les extensions actuelles sont :

Informations	**Extensions**
`ENTETE`	`.ent`
`ENTREE`	`.in`
`PROFIL`	`.lis`

C. LE PROGRAMME INFIERNO

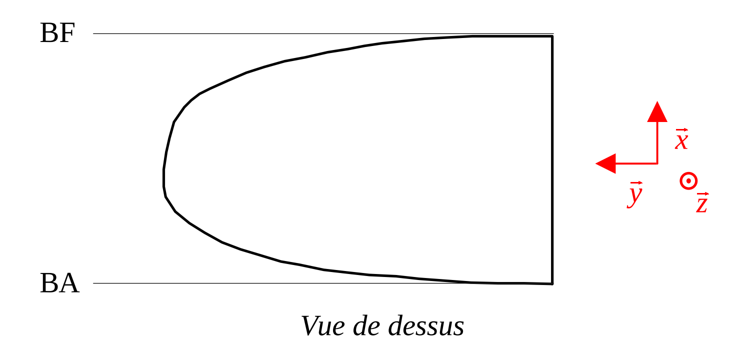
1. But

Ce programme occupe une place essentielle dans la chaîne précédemment présentée.
En effet, à partir de la description nervure par nervure d'un demi-parapente, il en génère un maillage complet. Ce maillage permet ensuite de plonger le demi-parapente obtenu dans une soufflerie numérique ou bien de le visualiser sous différents formats.
En plus de ce maillage, Infierno permet d'obtenir des informations sur les cellules du parapente ainsi que sur ses points caractéristiques.

2. Méthode de déploiement

Afin d'obtenir le maillage d'un parapente qui soit le plus réaliste possible, on monte le parapente nervure par nervure.

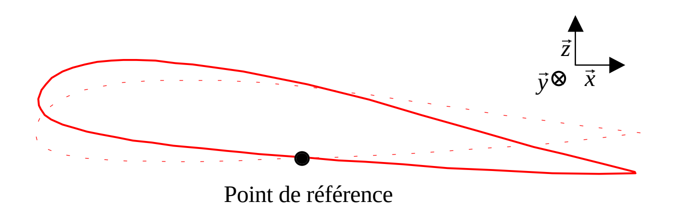
On choisit d'orienter le repère comme indiqué sur le schéma suivant :



Vue de dessus

Pour monter le parapente, on applique le schéma suivant :

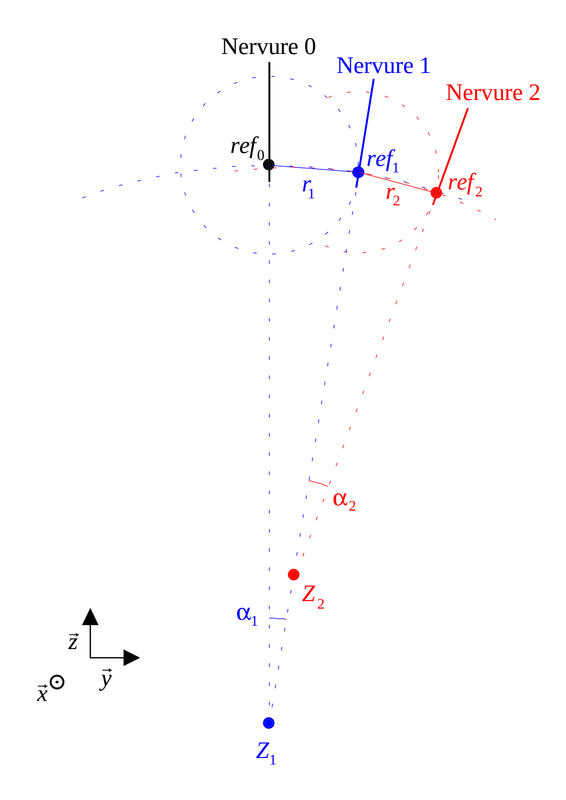
- On commence par placer le maillon au point de coordonnées $(0, demi_ecart_, 0)$.
- On superpose ensuite les points de référence de toutes les nervures à l'origine.
- On élève alors verticalement les nervures de façon à placer les points de référence en $(0, 0, suspentage_)$.
- On applique le vrillage à toutes les nervures.
  Cela consiste à calculer l'image de chaque nervure par la rotation ayant pour centre le point de référence, d'axe (Oy) et d'angle celui indiqué par la variable $vrillage_$.



- On déploie ensuite les nervures en envergure, c'est-à-dire que l'on applique aux nervures des rotations successives d'axe (Ox) mais d'angle et de centre à définir.
  La nervure 0 n'est pas déployée car il s' agit de la nervure centrale par convention.

Déployons la première nervure.
Le centre de la rotation est l'origine du repère.
Afin de déterminer l'angle de la rotation, on commence par placer le point de référence de la nervure 1 : il appartient aux cercles $C(ref_0, r_1)$ et $C(Z_1, R_0)$.
On a deux possibilités : on choisit le point d'intersection qui nous permet de déployer le parapente dans le sens qui nous intéresse.
Le calcul des coordonnées cartésiennes de cette intersection est réalisé par la méthode `Intersection` de la classe `Parapente`. Cette fonction utilise les formules obtenues symboliquement à l'aide de MAPLE.
Une fois le triangle (Z_1, ref_0, ref_1) mis en place, on calcule à l'aide de la formule d'Al Kashi l'angle α_1 de la rotation :

$$\cos\alpha_1 = \frac{\mathrm{d}(ref_0, ref_1)^2 - \mathrm{d}(ref_0, Z_1)^2 - \mathrm{d}(ref_1, Z_1)^2}{2\times\mathrm{d}(ref_0, Z_1)\times\mathrm{d}(ref_1, Z_1)}$$



Déployons la deuxième nervure.
On a besoin, comme précédemment, de déterminer le centre et l'angle de la rotation.
On note Z_2 le centre de la rotation : ce point appartient à la droite (Z_1, ref_1) et sa distance par rapport à ref_1 est donnée par la valeur du champ *rayon_* de la nervure 1.
On calcule l'angle α_2 de la même manière que α_1, c'est-à-dire en calculant ref_2 comme point d'intersection des cercles $C(ref_1, r2)$ et $C(Z_2, R_1)$, puis en utilisant la formule d'Al Kashi.

Il en va de même pour le déploiement des nervures restantes.

- On applique enfin le décalage des bords d'attaque qui consiste en une translation de chaque nervure par le vecteur de composantes $(decalBA_, 0, 0)$ où *decalBA_* est propre à chaque nervure.

3. Les formats de sortie et leur utilisation

a. Le format CMARC

Infierno peut générer un fichier de sortie au format CMARC qui est alors utilisable par la soufflerie numérique CMARC développée par Aerologic.

Ces fichiers sont constitués d'un en-tête initialisant un certain nombre de paramètres, d'une description des nervures (les sections) et se terminent par d'autres paramètres indiquant notamment les lignes de courant.

Pour plus d'informations, on pourra se reporter à la documentation disponible sur le site WEB d'Aerologic.

- **L'en-tête**

De manière générale, il est de la forme :

```
    Application Simul3D : Fichier d'entree pour CMARC
&BINP2   LSTINP=2,     LSTOUT=1,     LSTFRQ=0,     LENRUN=0,     LPLTYP=1,  &END
&BINP3   LSTGEO=2,     LSTNAB=0,     LSTWAK=0,     LSTCPV=0,                &END
&BINP4   MAXIT=250,    SOLRES=0.0005,                                       &END
&BINP5   NTSTPS=3,     DTSTEP=30,                                     &END
&BINP6   RSYM=0.0,     RGPR=0.0,     RFF=5.0,   RCORES=0.050,   RCOREW=0.050,&END
&BINP7   VINF=10,    VSOUND=300,                                      &END
&BINP8   ALDEG=0,    YAWDEG=0,   PHIDOT=0,   THEDOT=0,   PSIDOT=0,          &END
&BINP8A  PHIMAX=0.0, THEMAX=0.0,   PSIMAX=0.0,
        WRX=0.0,      WRY=0.0,      WRZ=0.0,                                &END
&BINP8B  DXMAX=0.0,    DYMAX=0.0,    DZMAX=0.0,
        WTX=0.0,      WTY=0.0,      WTZ=0.000,                              &END
&BINP9   CBAR=3,   SREF=28.7485,  SSPAN=6,
        RMPX=-1.5,    RMPY=0,    RMPZ=7.6008,                               &END
&BINP10  NORSET=0,     NBCHGE=0,     NCZONE=0,
        NCZPCH=0,     CZDUB=0.0,    VREF=0.0,                               &END
&BINP11  NORPCH=0,     NORF=0,       NORL=0,
        NOCF=0,       NOCL=0,       VNORM=0.0,                              &END
&BINP12  KPAN=0,       KSIDE=0,      NEWNAB=0,     NEWSID=0,                &END
&BINP13  NBLIT = 1,                                                   &END
&ASEM1   ASEMX=    0.0000, ASEMY=    0.0000, ASEMZ=    0.0000,
        ASCAL=    1.0000, ATHET=   0.0,      NODEA=   5,                    &END
&COMP1   COMPX=    0.0000, COMPY=    0.0000, COMPZ=    0.0000,
        CSCAL=    1.0000, CTHET=   0.0,      NODEC=   5,                    &END
```

Dans cet en-tête, seuls les champs de BINP9 sont spécifiques à chaque parapente, les autres champs sont prédéfinis et communs à tous les parapentes.

Valeurs et descriptions des paramètres retenus pour configurer la soufflerie numérique :

BINP2 : Output instructions

LSTINP	2	Print all input data
LSTOUT	1	Set additional output data print options below
LSTFREQ	0	Detailed panel data on last step, other data at each step
LENRUN	0	Complete run scope
LPLTYP	1	Formatted ASCII plot file

BINP3 : Print options

LSTGEO	2	Print all panel corners and unit normal vectors
LSTNAB	0	Panel neighbor information printout is off
LSTWAK	0	Wake data printout options is off
LSTCPV	0	Panel corner point analysis printout is off

BINP4 : Solver parameters

MAXIT	250	Limit on number of solver iterations
SOLRES	0.005	Convergence criteria for the matrix solver

BINP5 : Time step parameters

NTSTPS	3	Number of wake time-steps
DTSTEP	30	Size of the time-step (seconds)

BINP6 : Symetry and computation parameters

RSYM	0.0	Symetrical case about Y=0
RGPR	0.0	No ground plane modeled at Z=0
RFF	5.0	Far-field-factor
RCORES	0.05	Surface panel core radius
RCOREW	0.05	Wake panel core radius

BINP7 : Free stream conditions

VINF	10	Dimensional free stream velocity
VSOUND	300	Dimensional speed of sound

BINP8 : Angular position and rotation rates

ALDEG	0	Angle of attack in degrees
YAWDEG	0	Yaw angle in degrees
PHIDOT THEDOT PSIDOT	0	Rotation rate about global X, Y and Z axis (deg/sec)

BINP8A : Rotational oscillatory motion amplitudes and frequencies

PHIMAX THEMAX PSIMAX	0.0	Amplitude of oscil. rot. about X, Y and Z axis
WRX WRY WRZ	0.0	Frequency of oscil. rot. about X, Y and Z axis

BINP8B : Translational oscillatory motion aplitudes and frequencies

DXMAX DYMAX DZMAX	0.0	Amp. of oscil. translation about X, Y and Z axis
WTX WTY WTZ	0.0	Freq. of oscil. translation about X, Y and Z axis

BINP9 : Reference dimensions

CBAR	Reference chord used for normalizuing pitching moment
SREF	Reference area for force and moment coefficient
SSPAN	Semispan used for normalizing rolling and yawing moments
RMPX RMPY RMPZ	Coordinates of the moment reference point

Ainsi pour chaque parapente, on a :

CBAR	Valeur de la corde de la nervure centrale
SREF	Surface à plat du parapente ($2\times demi_surface_$)
SSPAN	Demi-envergure du parapente (*demi_env_)*
RPM	Coordonnées du BA de la nervure centrale

BINP10 : Special options

NORSET	0	The number of groups of panels on which nonzero normal velocities are to prescribed
NBCHGE	0	The number of panel neighbor information changes that are to be made
NCZONE	0	Regular external flow problem
NCZPCH	0	Patch number of the patch on which the value is specified for internal flow modeling
CZDUB	0.0	The value that is set on panel NCZPCH for internal flow modeling
VREF	0.0	VINF is used to compute Cp

`BINP11` : Normal velocity specification

`NORPCH`	0	Patch number of patch containing the group of panels to receive a prescribed normal velocity
`NORF`	0	All rows on this patch
`NOCF`	0	All columns on the patch
`VNORM`	0.0	Specified normal velocity for the set of panels identified above

`BINP12` : Panel neighbor information change

`KPAN`	0	Panel number
`KSIDE`	0	Side of the panel requiring a modified neighbor
`NEWNAB`	0	New neighbor
`NEWSID`	0	Side of that neighbor adjacent to `KSIDE` of `KSPAN`

`BINP13` : Boundary layer calculation control

`NBLIT`	1	Boundery layer computations performed on specified onbody streamlines

`ASEM1` : Assembly coordinate system information

`ASEMX ASEMY ASEMZ`	$(0,0,0)$	Origin of assembly coordinate system
`ASCAL`	1.0	Assembly scale
`ATHET`	0	Rotation angle of the assembly coordinate system about the rotation axis
`NODEA`	5	No other coordinate system

`COMP1` : Component coordinate system information

`COMPX COMPY COMPZ`	$(0,0,0)$	Origin of component coordinate system
`CSCAL`	1.0	Component scale
`CTHET`	0	Rotation angle of the component coordinate system about the rotation axis
`NODEC`	5	No other coordinate system

- **Le corps du fichier**

On rassemble dans `PATCH1` les nervures constituant le parapente.

```
&PATCH1 IREV=0, IDPAT=1, MAKE=0, KCOMP=1, KASS=1, IPATSYM=0, IPATCOP=0, &END
  Voile
```

Options retenues :

`IREV`	0	Patch not reversed
`IDPAT`	1	Wing type patch. Section force and moment data printed out
`MAKE`	0	Normal patch input
`KCOMP`	1	Number of component coordinate system
`KASS`	1	Number of assembly coordinate system
`IPATSYM`	0	No symetrical patch is generated
`IPATCOP`	0	

On décrit alors les nervures une par une :

```
&SECT1 STX=0.0, STY=0.0, STZ=0.0, SCALE=1.0, ALF=0.0, THETA=0.0, INMODE=4,
    TNODS=0, TNPS=0, TINTS=0, &END
1.5 0 7.6008
1.494 0 7.60119
...
1.494 0 7.60181
1.5 0 7.6008
&BPNODE TNODE=3, TNPC=0, TINTC=0, &END
```

```
&SECT1 STX=0.0, STY=0.0, STZ=0.0, SCALE=1.0, ALF=0.0, THETA=0.0, INMODE=4,
     TNODS=0, TNPS=0, TINTS=0, &END
 1.49848 0.333603 7.58784
 1.49249 0.333621 7.58825
 ...
```

Toutes les nervures ont en commun les champs suivants :

`SECT1` : Section coordinate system information

`STX STY STZ`	$(0, 0, 0)$	Origin of section coordinate system
`SCALE`	1.0	Section scale
`ALF`	0.0	Rotation angle of the section coordinate system about its Y axis
`THETA`	0.0	Rotation angle of the section coordinate system about its Z axis
`INMODE`	4	Input X, Y, Z coordinates to define
`TNODS`	0	First or intermediate section of patch
`TNPS`	0	No panel generated between this break point and the previous one
`TINTS`	0	Full cosine spacing of panels between this break point and the previous one

`BPNODE` : Break point input

`TNODE`	3	Final break point. End of this section definition
`TNPC`	0	No panel generated between this break point and the previous one
`TINTC`	0	Full cosine spacing of panels between this break point and the previous one

Seul le champ `TNODS` varie pour la dernière nervure : il vaut 3 et non 0, ce qui signifie que l'on termine la description de `PATCH1`.
La patch se termine par :

```
&PATCH1 IREV=0, IDPAT=1, MAKE=1, KCOMP=1, KASS=2, IPATSYM=0, IPATCOP=0, &END
```

La valeur de `MAKE` implique l'existence de `PATCH2`.

- **La fin du fichier**

On définit le `PATCH2` qui correspond aux stabilos.

```
&PATCH2 ITYP=1, TNODS=5, TNPS=0, TINTS=0,                              &END
..Stabilo
```

Options retenues :

`ITYP`	1	Flat tip patch
`TNODS`	5	Last patch in the surface geometry input
`TNPS`	0	No panel generated across the open tip
`TINTS`	0	Full cosine spacing of panels across the open tip

Suit alors la description du bord de fuite :

```
 &WAKE1   IDWAK=1,   IFLXW=0,    ITRFTZ=1,    INTRW=1,                 &END
  Bord de fuite
```

avec :

`IDWAK`	1	Regular wake
`IFLXW`	0	Flexible wake. Wake will be time-stepped with the local velocity
`ITRFTZ`	1	Wake separation line used for Trefftz plane computation of induce drag
`INTRW`	1	Intersection routine turned on

Puis les informations sur les séparations des WAKE :

```
&WAKE2   KWPACH=1,    KWSIDE=2,        KWLINE=0,       KWPAN1=0,
     KWPAN2=0,    NODEW=0,    INITIAL=0,                              &END
&WAKE2   KWPACH=2,    KWSIDE=1,        KWLINE=0,       KWPAN1=0,
     KWPAN2=0,    NODEW=5,    INITIAL=0,                              &END
```

avec :

KWPACH		Surface geometry patch number that this wake separates from
KWSIDE		Side of the patch which is parallel to separation line
KWLINE	0	Separation is from patch edge
KWPAN1	0	Defaults to the first panel on row or column
KWPAN2	0	Defaults to the last panel on row or column
NODEW		Indicates the next WAKE2
INITIAL	0	No initial wake geometry to be specified

On finit par les lignes de courant que l'on place arbitrairement toutes les trois nervures (à chaque fois, une ligne de courant sur l'intrados et une sur l'extrados) :

```
&ONSTRM  NONSL =12, KPSL = 20,60,260,300,500,540,740,780,980,1020,1220,1260   &END
&BLPARAM RN=2000000, VISC=0.000016, NSLBL=1,2,3,4,5,6,7,8,9,10,11,12  &END
```

Les valeurs des champs suivants varient selon chaque parapente :

ONSTRM :

NONSL		Number of on-body streamlines
KPSL		Surface panel number on wich each streamline starts
KWLINE	0	Separation is from patch edge
KWPAN1	0	Defaults to the first panel on row or column
KWPAN2	0	Defaults to the last panel on row or column
NODEW		Indicates the next WAKE2
INITIAL	0	No initial wake geometry to be specified

BLPARAM : Namelist for boundary layer parameters (needed NBLIT=1 on &BINP13)

RN	2000000	Reynolds number
VISC	0.000016	Dimensional kinematic viscosity
NSLBL		Streamline numbers of streamlines on which boundary layer calculations are to be performed

On indique enfin que l'on ne considère pas de « off-body velocity scan input section » :

```
&VS1     NVOLR= 0,       NVOLC= 0,                            &END
&SLIN1   NSTLIN=0,                                            &END
```

b. Les formats de visualisation pour le parapente

Afin d'avoir une idée de la forme du parapente que l'on a bâti, différents formats d'affichage sont prévus.

i. Le format GNUPLOT

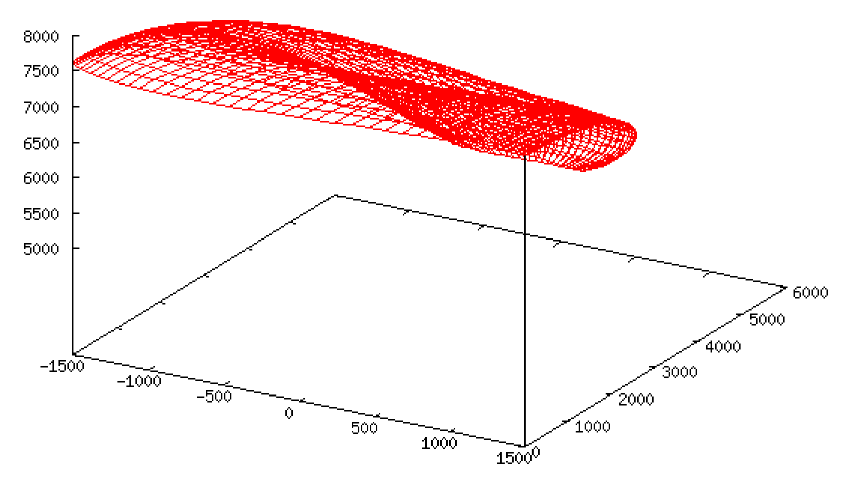
Comme son nom l'indique, il génère un fichier interprétable par le logiciel GNUPLOT.
Sa structure est assez naturelle :

```
#Parapente au format GNUPLOT

#Caracteristiques de la nervure
#------------------------------
#Fichier de points 2D : milieu.lis
#Nb de points : 80
#Corde    : 3000
#Largeur  : 0
#DecalBA  : 0
#Vrillage : 0
#Rayon    : 7500
#OuvEx    : 120
#OuvIn    : 360
#Points 3D de la nervure :
1500 0 7600.8
1494 0 7601.19
1482 0 7602.3
1458 0 7603.95
1428 0 7605.81
1386 0 7607.55
1338 0 7608.78
1278 0 7609.17
1215 0 7608.33
1140 0 7606.08
1062 0 7602.26
...
```

Ainsi, on rappelle les caractéristiques essentielles de chaque nervure puis on indique les coordonnées 3D des points constituant son contour.
Ces points sont listés en partant du BF, puis en suivant l'intrados pour atteindre le BA et pour finir en suivant l'extrados pour revenir au BF.
Pour chaque nervure, le BF est donc répété deux fois.

L'inconvénient majeur de cette interface est l'absence de possibilité de rotation de la figure obtenue.



Demi-parapente visualisé à l'aide de GNUPLOT

ii. Le format DXF

Ce format de sortie génère un fichier interprétable par AUTOCAD qui est un standard parmi les logiciels de CAO.
Le fichier est créé en faisant appel à GNUPLOT et en utilisant le terminal de sortie DXF.
On a donc besoin d'avoir en entrée un parapente au format GNUPLOT : si celui n'existe pas encore, on le crée automatiquement.

iii. Le format PLOTMTV

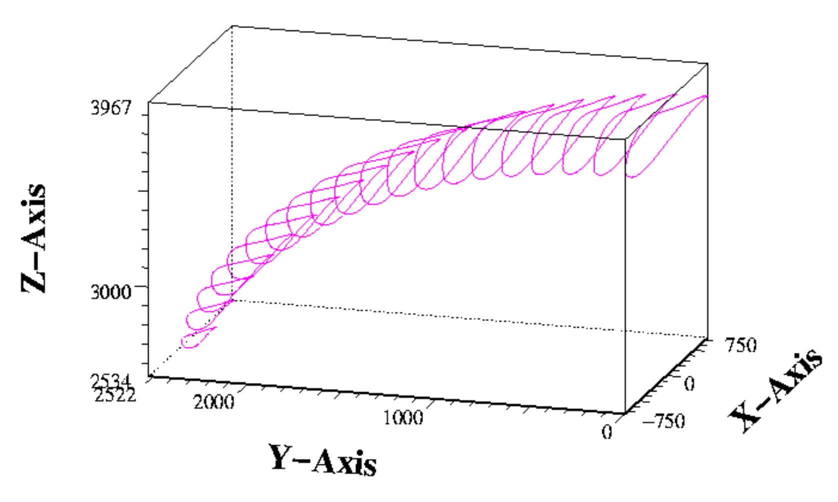
Il s'agit de créer un fichier interprétable par le viewer PLOTMTV développé par Kenny K.H Toh.
Sa structure est analogue à celle d'un fichier au format GNUPLOT :

```
#Parapente au format PLOTMTV
$ DATA=CURVE3D
% axisscale=False

#Caracteristiques de la nervure
#-------------------------------
#Fichier de points 2D : milieu.lis
#Nb de points : 80
#Corde    : 3000
#Largeur  : 0
#DecalBA  : 0
#Vrillage : 0
#Rayon    : 7500
#OuvEx    : 120
#OuvIn    : 360
#Points 3D de la nervure :
1500 0 7600.8
1494 0 7601.19
1482 0 7602.3
1458 0 7603.95
1428 0 7605.81
1386 0 7607.55
...
```

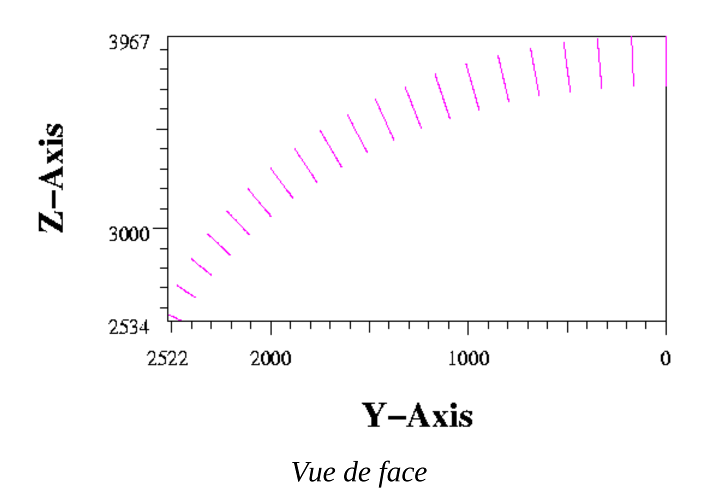
La seule différence réside dans l'en-tête du fichier où l'on indique que l'on fournit des données en 3D et que l'on souhaite avoir un repère orthonormé.

Cette interface présente l'énorme avantage de pouvoir faire tourner dans toutes les directions le graphe tracé.
Cependant, on obtient une figure filaire et non pas un volume :



Vue de trois quart

On peut quand même voir l'influence du rayon utilisé pour le déploiement, c'es t-à-dire le paramètre qui sert à arquer plus ou moins le parapente :



Vue de face

iv. Le format MATLAB

Ce format a été développé pour permettre l'affichage du parapente bâti grâce à l'interface graphique de MATLAB. Il permet de représenter de véritables volumes et de générer des fichiers au format VRML sans trop de complications.
Cette sortie présente cependant deux inconvénients : elle n'est utilisable que lorsqu'on travaille sous Windows et elle nécessite en plus un programme pour interpréter le fichier contenant le maillage du parapente.

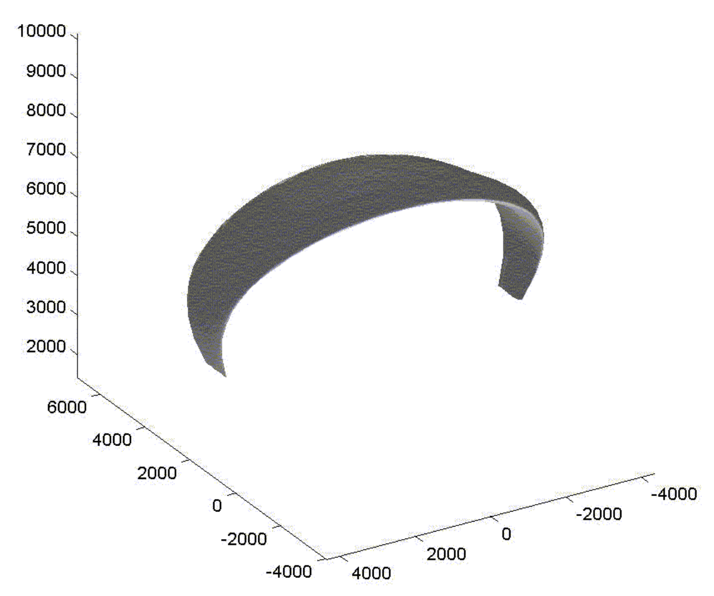
Format d'un tel fichier :

```
20  81  0
469.43 4929.69 5084.01
467.569 4929.9 5084.15
463.855 4930.41 5084.47
456.416 4931.29 5085.02
447.112 4932.34 5085.68
434.067 4933.57 5086.46
419.142 4934.78 5087.22
400.461 4935.99 5087.98
380.819 4936.93 5088.57
357.408 4937.72 5089.06
...
```

La première ligne indique le nombre de nervures formant le parapente ainsi que le nombre de points par nervure.
Suivent alors les coordonnées de tous les points du maillage en partant de l'extrémité de l'aile pour rejoindre la nervure centrale.

Ce format de fichier est interprété par le programme MATLAB nommé **afentier** qui génère automatiquement un fichier VRML.

Nous avons également créé, à titre d'exemple, un programme nommé **mtb2mtb_full** qui effectue la symétrie de la demi-aile, créant ainsi le parapente complet.



Parapente complet créé avec MATLAB

Lors de nos recherches sur le WEB, nous avons trouvé un applet Java permettant de visualiser des fichiers WAVEFRONT .obj mais aussi et surtout de les faire tourner facilement à la souris. De plus, le fait que ce soit un applet Java permet un transport vers des plates-formes autres que Linux.

Pour utiliser cet applet, nous avons créé un programme **mtb2obj_full** qui convertit un fichier au format MATLAB en un fichier au format Wavefront. Tout comme mtb2mtb_full, ce programme génère le parapente complet.

On décrit tout d'abord les sommets du maillage, dits « vertex » :

```
v 469.43  4929.69  5084.01
v 467.569  4929.9  5084.15
v 463.855  4930.41  5084.47
v 456.416  4931.29  5085.02
...
```

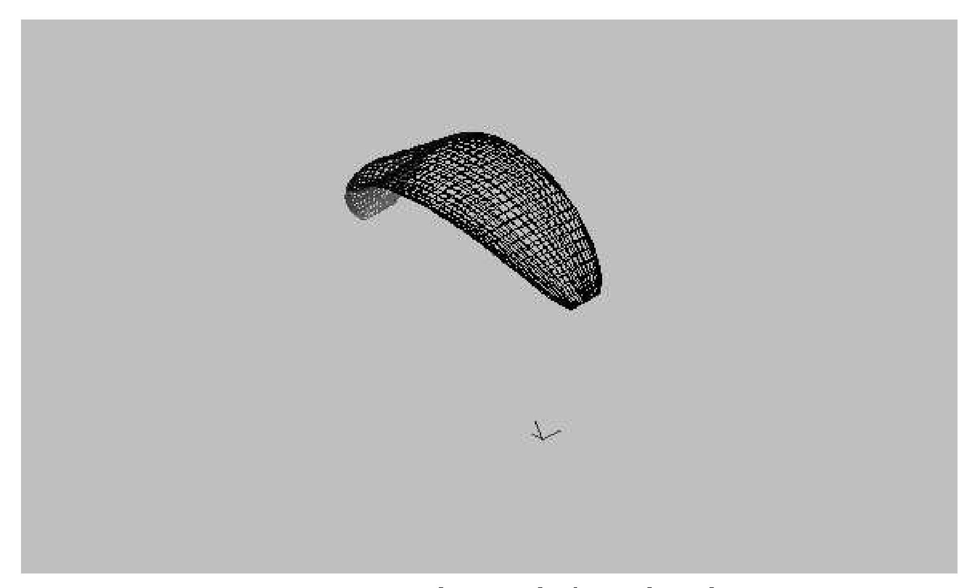
Dans la suite du fichier, la description des faces et des lignes se sert des numéros des vertex dans l'ordre où ils ont été énumérés.

On décrit ensuite les faces, c'est-à-dire les nervures :

```
f 1  2  3  4  5  6  7  8  9  10  11  12  13  14...
f 15  16  17  18...
...
```

Enfin les lignes intermédiaires :

```
l 21  102  183  264  345  426  507  588...
l 22  103  184  265  346  427  508  589...
...
```



Parapente complet visualisé avec l'applet

*Voir en annexe les syntaxes d'appel de **afentier**, **mtb2mtb_full** et **mtb2obj_full**.*

c. Les informations complémentaires

i. *La sortie* CELLULE

Cette sortie rassemble dans un fichier les informations calculées pour chaque cellule qui est l'espace délimité par deux nervures consécutives.

Pour fixer les idées, voici un extrait d'un tel fichier :

```
#Cellule 1
Corde              (en m) = 2.9906
Largeur            (en m) = 0.330807
Surface          (en m^2) = 0.989168
Surface Mouillee (en m^2) = 2.05982
Volume          (en dm^3) = 280.039
Rayon de bord d'attaque   = 1.49231
Bord d'attaque            = -1494.72 ; 499.431 ; 7590.4
Bord de fuite             = 1495.83 ; 498.217 ; 7573.97
```

Sauf indication contraire, on considère dans ce qui suit la cellule formée par les nervures i et $i+1$.
Pour cette cellule, on calcule les éléments caractéristiques suivants :

- **La corde**

Il s'agit de la corde moyenne de la cellule :

$$corde_cellule_ = \frac{\text{corde}_i + \text{corde}_{i+1}}{2}$$

- **La largeur**

Il s'agit de la largeur moyenne de la cellule obtenue en utilisant la largeur au niveau des BA et celle au niveau des BF :

$$largeur_cellule_ = \frac{d(BA_i, BA_{i+1}) + d(BF_i, BF_{i+1})}{2}$$

- **La surface**

Il s'agit de la surface à plat de la cellule, c'est -à-dire de la surface entre les cordes des nervures attenantes.
Elle est calculée en sommant la surface des deux triangles :

$$surface_cellule_ = \text{surface}(BA_i, BA_{i+1}, BF_i) + \text{surface}(BA_{i+1}, BF_{i+1}, BF_i)$$

- **La surface mouillée**

On appelle ainsi la surface extérieure de la cellule. Pour la calculer, on somme les surfaces des triangles formés par les points des contours des deux nervures :

$$surface_mouillée_ = \sum_{j=0}^{n-1} \left[\text{surface}(\text{ner}_i(j), \text{ner}_i(j+1), \text{ner}_{i+1}(j+1)) + \text{surface}(\text{ner}_i(j), \text{ner}_{i+1}(j), \text{ner}_{i+1}(j+1))\right]$$

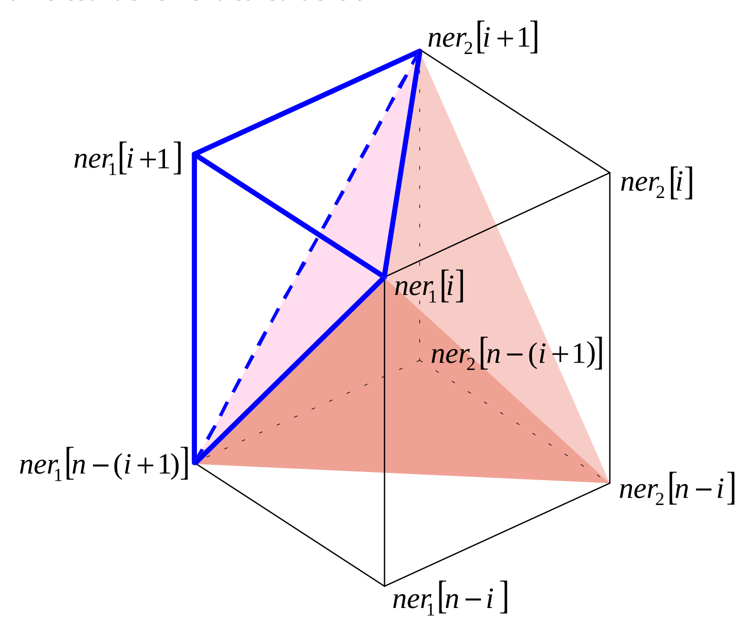
- **Le volume**

Comme son nom l'indique, il s'agit du volume de la cellule.
Pour ne pas alourdir les notations, on considère que les nervures définissant la cellule sont les nervures 1 et 2.
Pour calculer le volume de cette cellule, on la découpe en tranches définies par les points :

$$ner_1[i],\ ner_1[i+1],\ ner_1[n-(i+1)],\ ner_1[n-i],\ ner_2[i],\ ner_2[i+1],\ ner_2[n-(i+1)] \text{ et } ner_2[n-i].$$

Un tel découpage n'est possible que si l'on a le même nombre de points sur l'intrados que sur l'extrados.

Afin d'approcher au mieux le volume de chacune de ces tranches, on les découpe à leur tour en tétraèdres dont le volume est facilement calculable :



Comme on le voit sur la figure, on obtient 5 tétraèdres dont 4 ont le même volume :

- $ner_1[i]$, $ner_1[i+1]$, $ner_1[n-(i+1)]$, $ner_2[i]$
- $ner_1[i]$, $ner_2[i]$, $ner_2[i+1]$, $ner_2[n-i]$
- $ner_1[i]$, $ner_1[n-i]$, $ner_1[n-(i+1)]$, $ner_2[n-i]$
- $ner_1[n-(i+1)]$, $ner_2[i+1]$, $ner_2[n-(i+1)]$, $ner_2[n-i]$
- $ner_1[i]$, $ner_1[n-(i+1)]$, $ner_2[i+1]$, $ner_2[n-i]$

- **Le bord d'attaque et le bord de fuite**

Il s'agit des coordonnées de BA et de BF moyens, en ce sens que :

$$BA = \frac{BA_i + BA_{i+1}}{2} \quad \text{et} \quad BF = \frac{BF_i + BF_{i+1}}{2}$$

- **Le rayon du bord d'attaque**

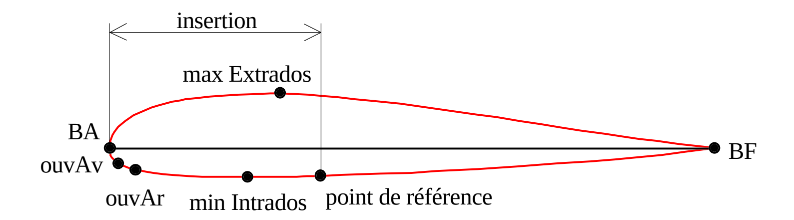
Il correspond au rayon du cercle circonscrit au BA de la cellule et aux deux points qui l'entourent.
Pour le calculer, on fait appel à la méthode `Rayon_Cercle_Circonscrit` de la classe `Point3D`.

Remarque :

Pour faciliter la lecture des valeurs calculées, on donne les longueurs en mètres, les surfaces en m^2 et les volumes en dm^3.

ii. La sortie POINT_R

On rappelle que pour chaque nervure, les points remarquables sont :



Les BA et BF de chaque nervure étant facilement identifiables dans les autres formats de sortie, on rassemble uniquement les coordonnées des ouvertures avant et arrière, des points haut et bas de la nervure ainsi que celles du point de référence dans le même fichier.
On ajoute également dans ce fichier le rayon du cercle circonscrit au point de référence de la nervure considérée et à ceux des deux nervures adjacentes.
L'intérêt d'un tel fichier est de pouvoir visualiser à l'a ide de PLOTMTV les différents points remarquables. On peut également les superposer au tracé du parapente à proprement parler.

Un exemple est suffisant pour avoir une idée de la structure de ce fichier :

```
#Fichier des points remarquables au format PLOTMTV
$ DATA=CURVE3D
% axisscale=False
% dlinetype=0
#Ouverture_Avant
% linelabel='Ouvertures'
% mt=6 mc=7
-1380  0  7517.31
-1378.2  330.404  7515.09
...

#Ouverture_Arriere
% linelabel='Ouvertures'
% mt=6 mc=7
-1140  0  7453.8
-1138.74  327.58  7450.84
...

#Maximum_Extrados
% linelabel='Max Extrados'
% mt=4 mc=2
-783  0  7933.8
-780.424  348.579  7928.47
...

#Minimum_Intrados
% linelabel='Min Intrados'
% mt=4 mc=5
-882  0  7434.6
-881.132  326.696  7430.74
...

#Point_Reference
% linelabel='Pt référence'
% mt=12 mc=4
0  0  7500
0  329.422  7492.76
...

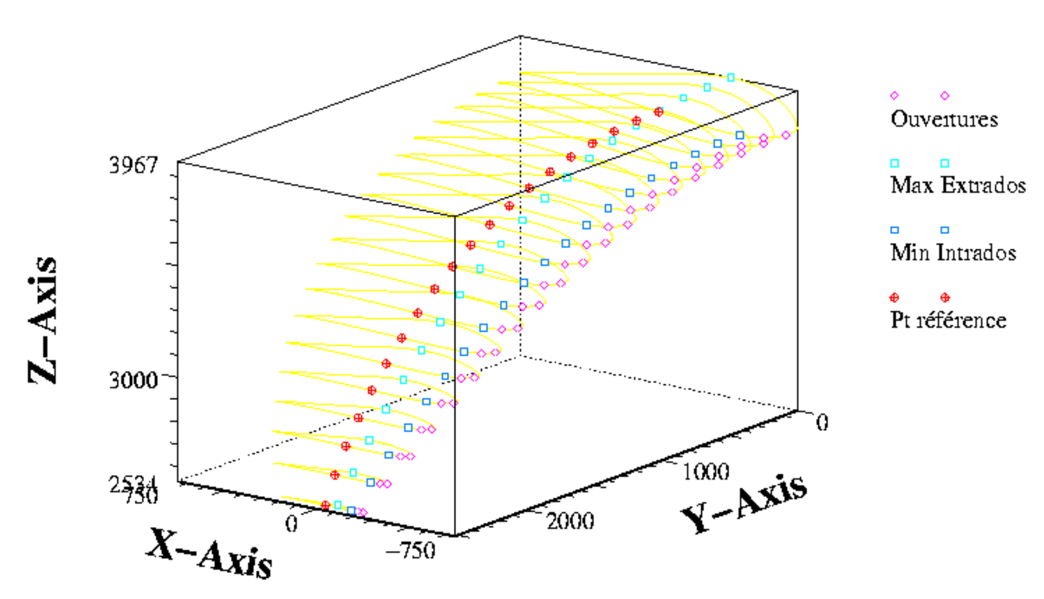
#Rayon_Courbure
# Nervure 0  7500
# Nervure 1  7386.35
...
```

Le fichier devant être interprétable par PLOTMTV, les lignes de l'en-tête indiquent que l'on fournit des données en 3D, que l'on désire avoir un repère orthonormé et que toutes les données qui suivent doivent être interprétées comme des points isolés et non pas comme des points appartenant à une même droite.

Suivent alors les coordonnées des points correspondant à toutes les ouvertures avant qui seront représentés par des marqueurs de type `mt` et de couleur `mc`.
Puis sur le même schéma on trouve les ouvertures arrière, les points haut et bas de la nervure et enfin les points de référence.
Pour finir, on indique en commentaires les rayons de courbure pour chaque nervure avec comme convention :

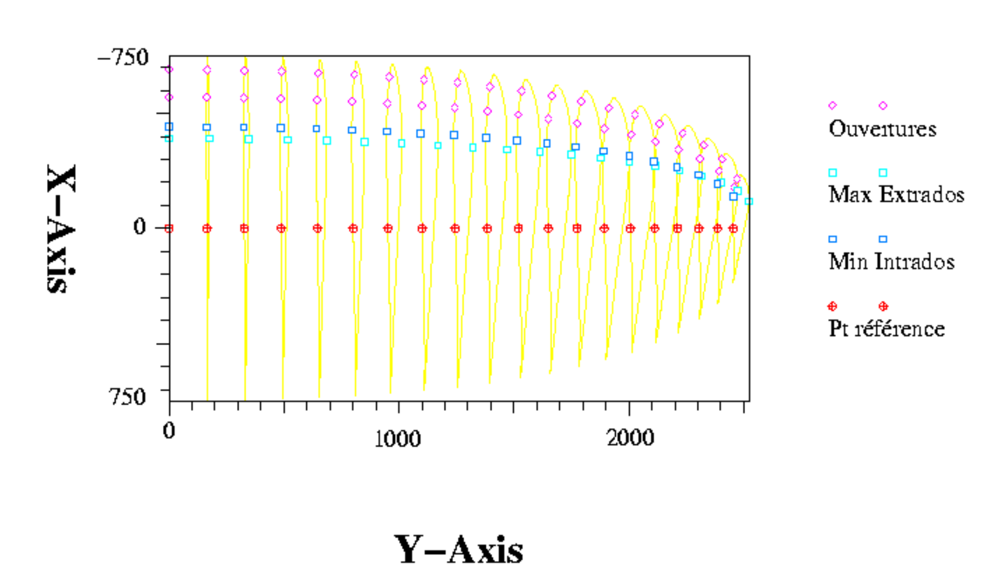
- 0 pour la dernière nervure
- -1 pour un rayon infini

Voici un exemple de visualisation dans PLOTMTV :



Demi-parapente avec ses points caractéristiques

Par ailleurs, PLOTMTV nous permet de vérifier que les points de référence sont alignés :



Vue de dessus d'un demi-parapente

iii. La sortie SUSPENTE

Cette sortie permet de rassembler dans un seul fichier les coordonnées des points d'ancrage et les longueurs des suspentes (distance du point d'ancrage au maillon) et ce pour chaque nervure.

Comme pour les points remarquables, le fichier obtenu est interprétable par PLOTMTV, ce qui permet de visualiser les points d'ancrage.

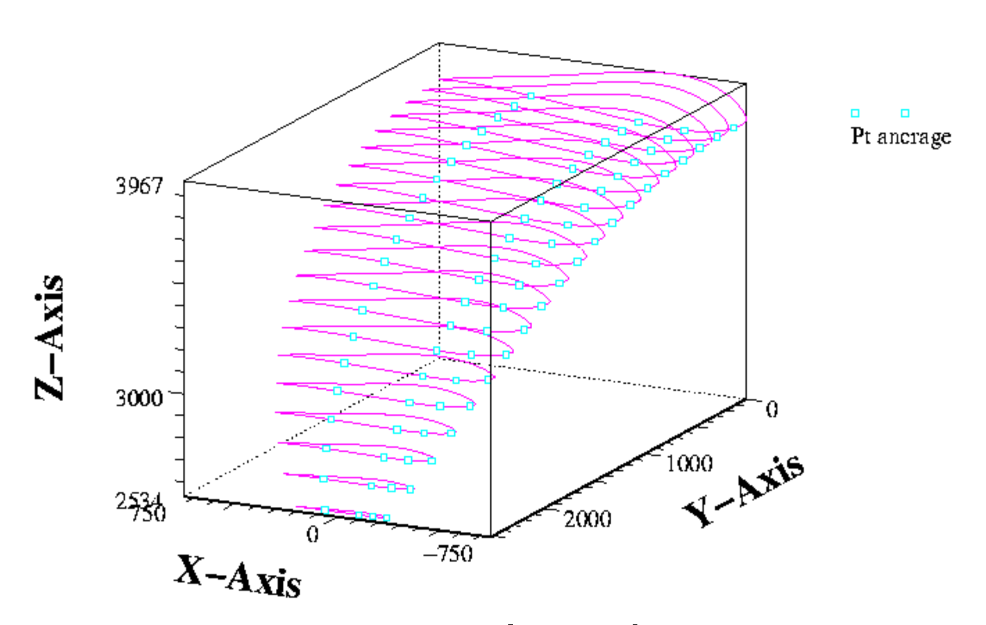
Structure du fichier :

```
#Fichier des suspentes au format PLOTMTV
$ DATA=CURVE3D
% axisscale=False
% dlinelabel='Pt ancrage'
% dlinetype=0 dmarkercolor=2 dmarkertype=4

#Nervure 0
#Points d'ancrage - Longueur des suspentes
-1380  0  7517.31
-900  0  7435.12
-450  0  7454.84
600  0  7564.12
#7645.54  7492.07  7471.09  7590.52

#Nervure 1
#Points d'ancrage - Longueur des suspentes
-1378.2  330.404  7515.09
-899.107  326.722  7431.33
...
```

On obtient à l'aide de PLOTMTV des représentations similaires à :



Demi-parapente et points d'ancrage des suspentes

4. La syntaxe d'appel

La syntaxe générale d'appel du programme Infierno est de la forme :

`Infierno -i fic_in [-e fic_ent] [-f format] [-d] [-m info]`

Détail des options :

`-i` Fichier d'entrée au format généré par le programme Corde.
Cette option est indispensable et indique à Infierno la description détaillée du parapente.

`-e` Fichier en-tête.
On utilise cette option si l'on ne possède qu'une description sommaire du parapente.
Infierno fait alors appel à Corde pour générer une description détaillée qui est stockée dans le fichier indiqué après la balise `-i`. L'utilisateur doit cependant indiquer pour chaque nervure le profil utilisé.

`-f` Format de sortie désiré.
Le champ `format` peut prendre les valeurs suivantes :

`GNUPLOT`, `PLOTMTV`, `LIBRE`, `CMARC`, `DXF` et `MATLAB`

chacun de ces formats ayant été détaillé précédemment.
Remarque : le format par défaut est `CMARC`.

`-d` Affichage automatique au format demandé.
Cette option permet l'affichage automatique du parapente uniquement dans GNUPLOT ou PLOTMTV.
Elle permet également de visualiser automatiquement les points remarquables et les points d'ancrage des suspentes si ces sorties sont demandées.

`-m` Informations complémentaires.
Le champ `info` peut prendre les valeurs suivantes :

`POINT_R`, `SUSPENTE` et `CELLULE`

On peut se reporter à la partie précédente pour une présentation détaillée de ces sorties.

Remarque : si l'option `-d` est activée, les points remarquables et les points d'ancrage des suspentes sont visualisés dans PLOTMTV.
De plus, si l'utilisateur a aussi demandé une sortie au format `PLOTMTV` pour le parapente, alors on superpose les informations complémentaires au tracé du parapente.

En cas de problèmes, une aide est proposée par la commande : **`Infierno –h`**

Par ailleurs, les extensions des fichiers générés pour toutes ces sorties sont indiquées dans le fichier `constantes.h`.
Ainsi, les extensions actuelles sont :

Formats	Extensions
CMARC	.cmi
GNUPLOT	.gnu
DXF	.dxf
LIBRE	.fre
MATLAB	.mtb
PLOTMTV	.mtv

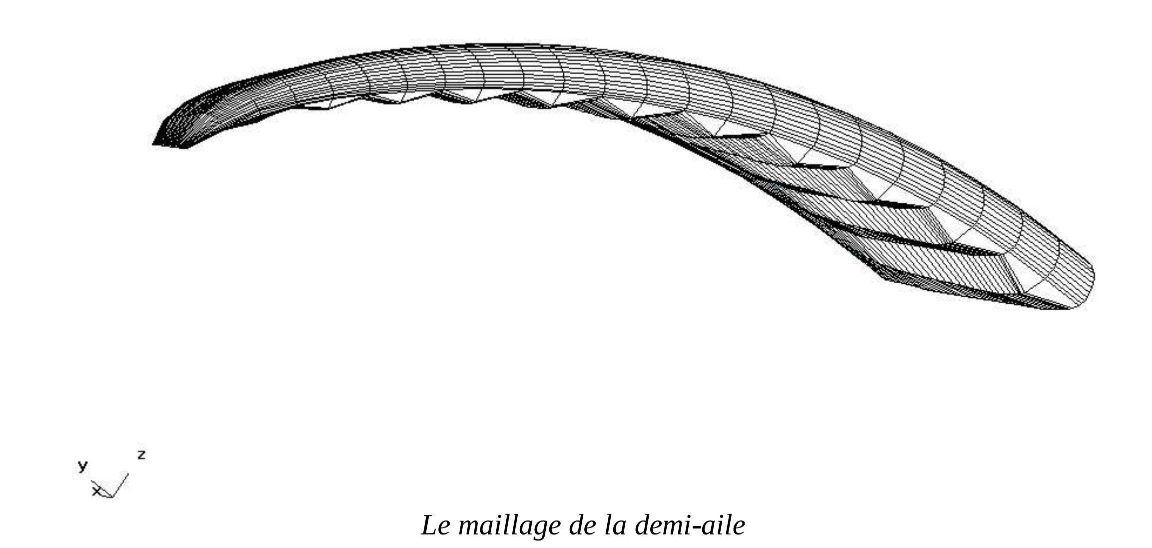
Informations	Extensions
CELLULE	.cmi
POINT_R	.gnu
SUSPENTE	.lib

IV. UTILISATIONS POSSIBLES AVEC CMARC ET HIPPOLYTE

La soufflerie numérique CMARC génère à partir d'un fichier .cmi, des fichiers .cm0, .out et .fmt contenant les résultats des essais dans la soufflerie numérique.
En visitant le site des concepteurs de CMARC, nous avons découvert que ceux-ci proposent POSTMARC, un post-processeur de cette soufflerie, permettant de visualiser le maillage de l'objet soufflé ainsi que les pressions qui s'exercent sur celui-ci.
Sans conviction, nous avons alors pris l'initiative d'envoyer quelques parapentes au format .cmi aux USA.
Le charme pyrénéen a agi et de superbes vues nous ont été renvoyées, accompagnées de quelques mots sympathiques sur les Pyrénées. Cela nous a permis de vérifier la validité des fichiers .cmi construits.

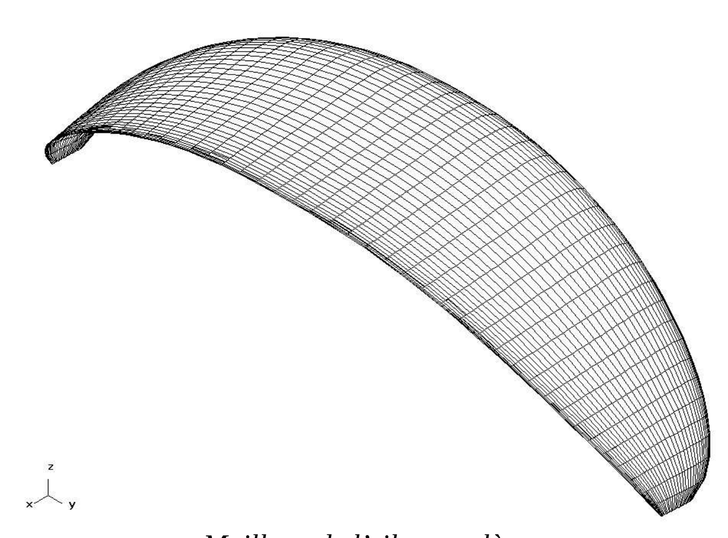
Voici donc ces *e*-résultats pour les prototypes Metpat et Kenya50.

- **Le prototype Metpat :**



Le maillage de la demi-aile

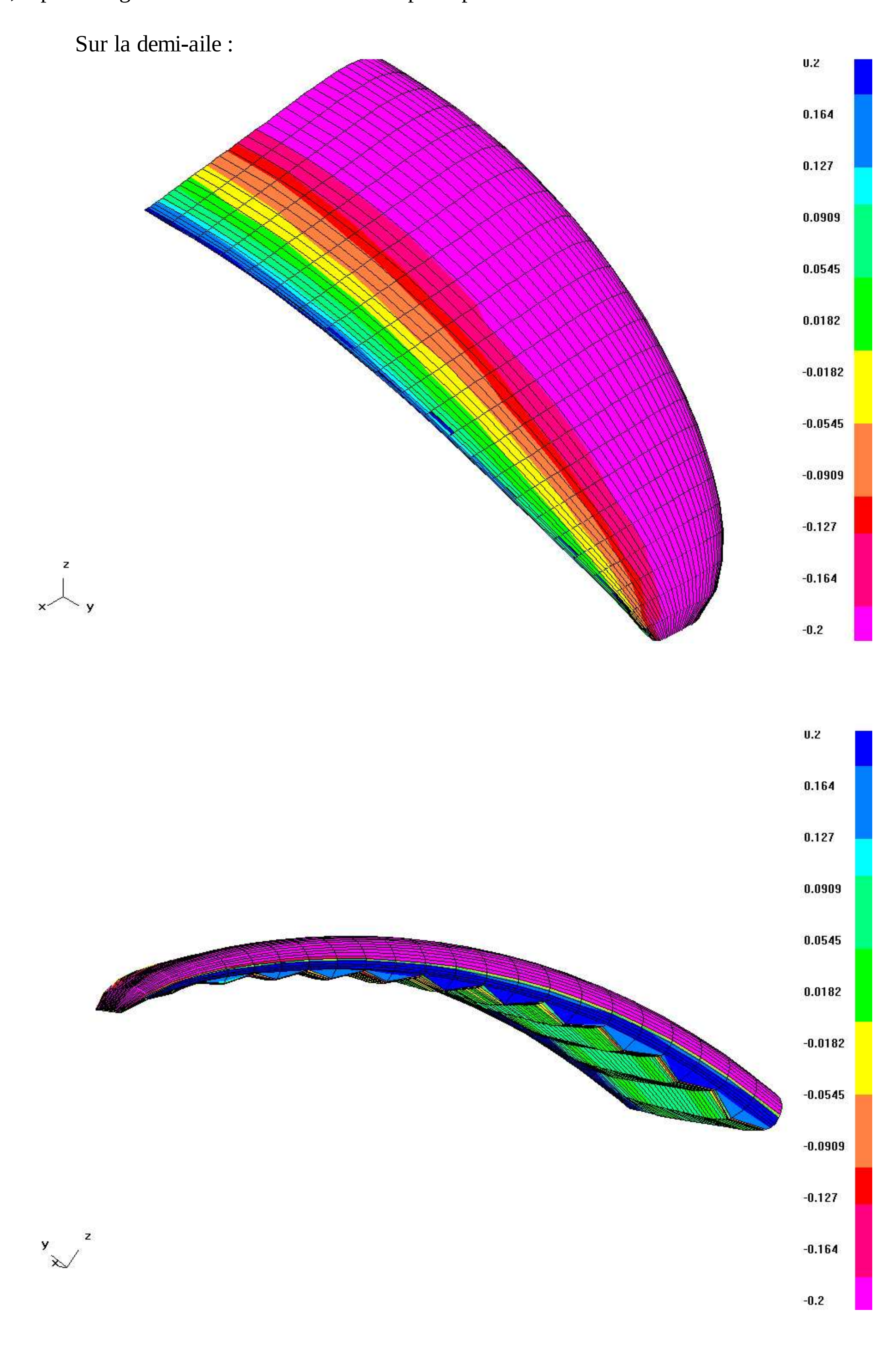
POSTMARC permet de symétriser le maillage :



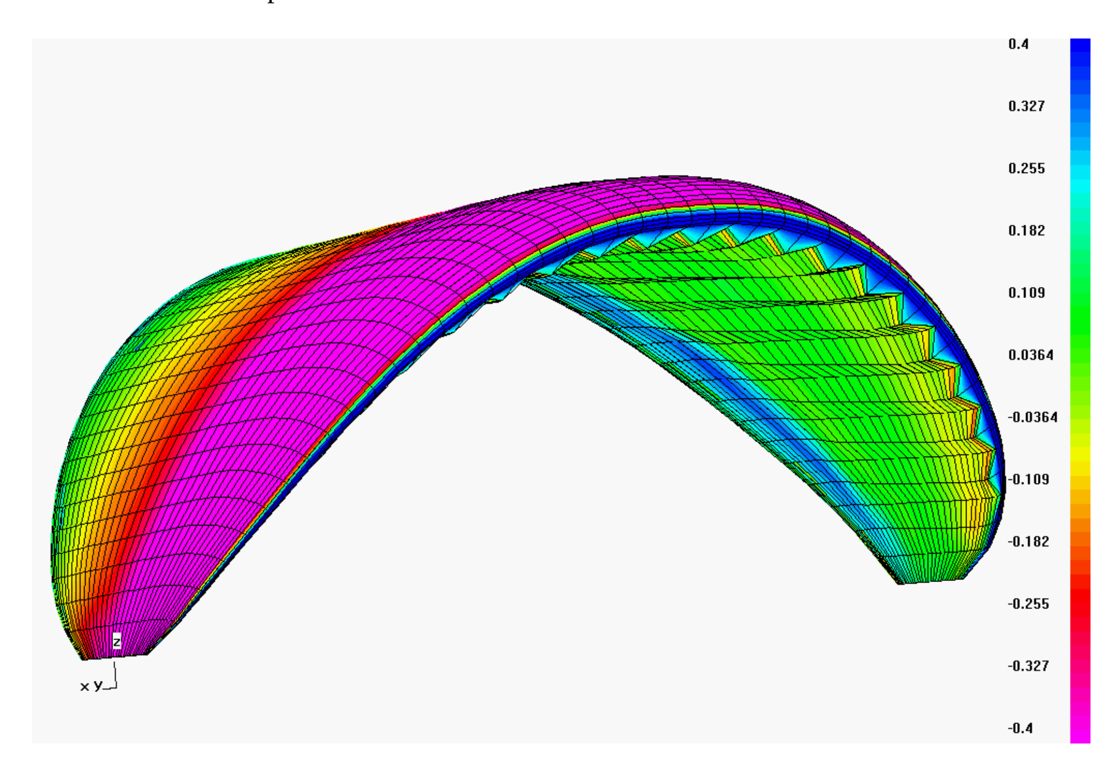
Maillage de l'aile complète

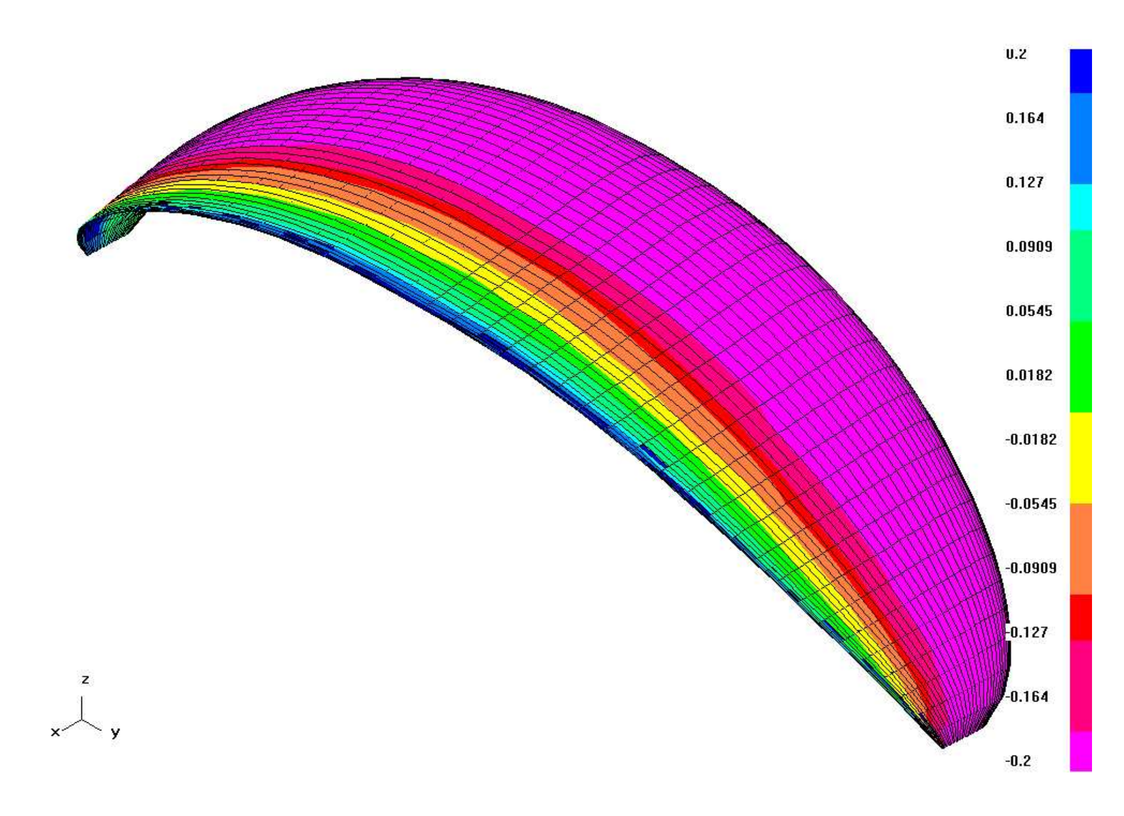
Mais, il permet également de visualiser les champs de pression.

Sur la demi-aile :

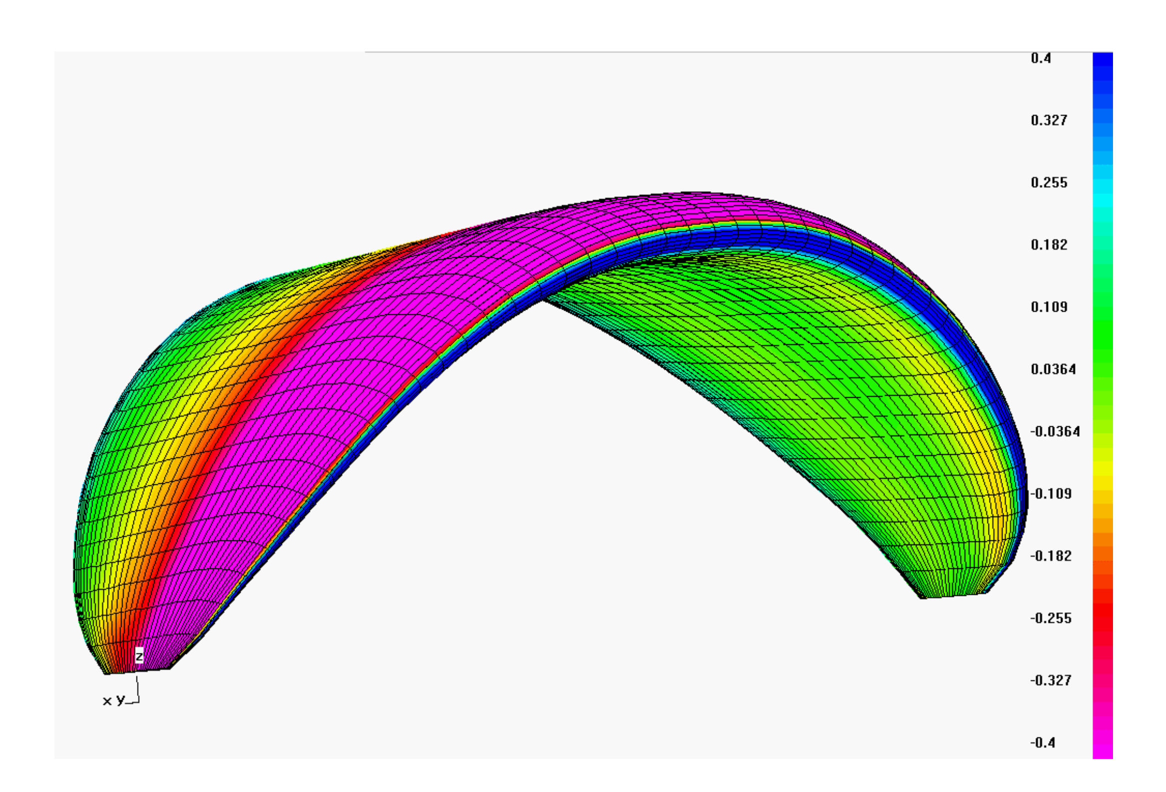


Sur l’aile complète :

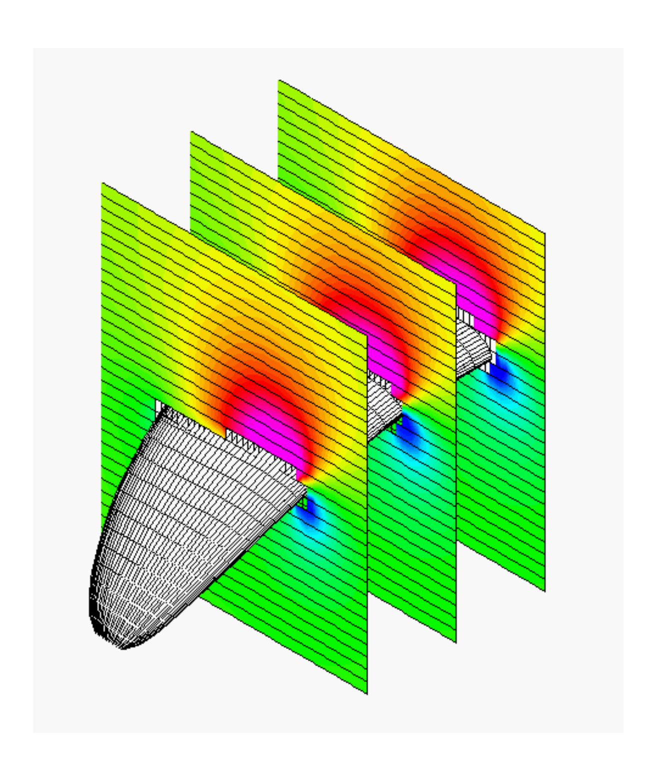




- **Le prototype Kenya50 :**



POSTMARC propose également de visualiser les champs de pression selon divers plans de coupe :



V. CONCLUSION

La chaîne Corde - Infierno permet donc, à partir d'un profil et d'une description sommaire de l'objet final, de générer le maillage de l'objet complet.
Le but était ici le maillage d'un parapente, mais les concepts utilisés resteraient valables pour d'autres structures utilisant un profil simple telles qu'un Kayak ou un dirigeable Zeppelin par exemple.

D'un point de vue personnel, ce projet nous a permis de mesurer la difficulté de s'insérer dans une chaîne de production déjà en place. Nous avons dû nous adapter aux contraintes des divers formats de fichier, aux objets déjà développés en Pascal…

Ce projet nous a également fait découvrir le travail en équipe et surtout le télétravail : nous étions deux étudiants à développer Corde et Infierno, mais nous n'avions que très rarement l'occasion de nous rencontrer. La magie d'Internet aidant, nous avons pu mettre chaque jour nos travaux en commun.

Nous avons aussi été amenés à rechercher et à utiliser des outils de développements adaptés comme XDIFF pour la gestion des versions, PLOTMTV pour un affichage plus agréable ainsi que DOXYGEN et A2PS pour la documentation.

Enfin, le choix de la plate-forme Linux nous a permis de découvrir la gestion de ce système et de nous perfectionner dans cet environnement plein d'avenir.

Conscients de l'utilité du projet global pour l'industrie du vol libre, il ne nous reste plus qu'à lui souhaiter « bon vent » …

ANNEXES

a. Le programme de visualisation MATLAB

Afentier permet de visualiser un parapente sous MATLAB et de générer un fichier VRML

```
% Le fichier d'entrée provient de la sortie MATLAB d'Infierno :
%     il contient un demi-parapente.
%
% Structure du fichier
% ligne 1 :       nombre de nervures de la demi-aile, nombre de points par nervure, 0
% ligne 2 à fin : les nervures en 3D stockées sous forme de coordonnées x y z
%                 en partant de l'extrémité de la demi-aile jusqu'à la nervure centrale
%                 et en énumérant les points dans l'ordre BF->intrados->BA->extrados->BF
%
% En sortie, on obtient :
%    * la représentation du parapente complet
%    * le fichier correspondant au format VRML

function afentier(nom)

clf  % On efface les figures

% On stocke dans une matrice tous les points du maillage
fid=fopen(nom,'r');
a=fscanf(fid,'%f %f %f',[3 inf]);
fclose(fid);
a=a';

% On récupère le nombre de nervures et le nombre de points par nervure
nn=floor(a(1,1));
npn=floor(a(1,2));
n=nn*npn;
a(1,:)=[];

% On isole x, y et z
x=a(:,1);
y=a(:,2);
z=a(:,3);
clear a;

% On ajoute le symétrique
for i=0:nn-1
   x(n+1+i*npn:n+(i+1)*npn)=x(n+1-(i+1)*npn:n-i*npn);
   y(n+1+i*npn:n+(i+1)*npn)=-y(n+1-(i+1)*npn:n-i*npn);
   z(n+1+i*npn:n+(i+1)*npn)=z(n+1-(i+1)*npn:n-i*npn);
end

% Désormais on a le double de nervures
nn=nn*2;
n=nn*npn;

% On va créer deux matrices :
%    * l'une contenant les coordonnées des points de l'intrados
%    * l'autre ceux de l'extrados
% ce qui permet de tracer ensuite l'intrados et/ou l'extrados

% On isole les points de l'intrados
for i=0:nn-1
   xi(1+i*((npn-1)/2+1):(i+1)*((npn-1)/2+1))=x(1+i*npn:(npn-1)/2+1+i*npn);
   yi(1+i*((npn-1)/2+1):(i+1)*((npn-1)/2+1))=y(1+i*npn:(npn-1)/2+1+i*npn);
   zi(1+i*((npn-1)/2+1):(i+1)*((npn-1)/2+1))=z(1+i*npn:(npn-1)/2+1+i*npn);
end;
% On fait une matrice de xi yi et zi
for i=0:(nn-1)
   for j=1:((npn-1)/2+1)
```

```
      xxi(i+1,j)=xi(i*((npn-1)/2+1)+j);
      yyi(i+1,j)=yi(i*((npn-1)/2+1)+j);
      zzi(i+1,j)=zi(i*((npn-1)/2+1)+j);
   end
end
clear xi;
clear yi;
clear zi;

% On isole les points de l'extrados
for i=0:nn-1
   xe(1+i*((npn-1)/2+1):(i+1)*((npn-1)/2+1))=x((npn-1)/2+1+i*npn:(i+1)*npn);
   ye(1+i*((npn-1)/2+1):(i+1)*((npn-1)/2+1))=y((npn-1)/2+1+i*npn:(i+1)*npn);
   ze(1+i*((npn-1)/2+1):(i+1)*((npn-1)/2+1))=z((npn-1)/2+1+i*npn:(i+1)*npn);
end;
% On fait une matrice de xe ye et ze
for i=0:(nn-1)
   for j=1:((npn-1)/2+1)
      xxe(i+1,j)=xe(i*((npn-1)/2+1)+j);
      yye(i+1,j)=ye(i*((npn-1)/2+1)+j);
      zze(i+1,j)=ze(i*((npn-1)/2+1)+j);
   end
end
clear xe;
clear ye;
clear ze;

% Tracé du parapente à proprement parler et génération du fichier au format VRML
figure(1)
hold on
axis equal
surfl(xxi,yyi,zzi)
surfl(xxe,yye,zze)
colormap gray
shading interp

% Détermination du nom du fichier au format VRML
i=0;
while nom(i+1)~='.'
   i=i+1;
end
nom=[nom(1:i),'.wrl']

vrml(figure(1),nom)
```

b. Syntaxe d'appel de mtb2obj_full

Deux syntaxes d'appel sont possibles :

mtb2obj_full
mtb2obj_full [fic_in]

Le premier appel, demande le nom du fichier à convertir alors que le deuxième appel prend ce nom en paramètre. Le fichier converti est alors systématiquement appelé parapente.obj.
Dans les deux appels, le navigateur Netscape est lancé automatiquement pour visualiser, à l'aide de l'applet, le parapente complet.

BIBLIOGRAPHIE ET SITES WEB

Concernant le parapente :

- *La folle histoire du parapente*, Xavier Murillo
- *Volez en parapente*, Gérald Delorme
- *Modélisation d'un parapente*, Jérôme Sarthe, rapport de stage de DESS - Université de Savoie 1999

- Manuel du parapente sur le site de la Fédération Française de Parachute :
  http://www.sxb.rte.fr/FFP-manuel/
- Site de la Fédération Française de Vol Libre :
  http://www.ffvl.fr/Vol_Libre/Historique/historique.html
- Article concernant la simulation numérique sur le site de P. Puiseux :
  http://www.univ-pau.fr/~puiseux/simulation/article/article.htm

Concernant la programmation et les logiciels :

- *Langage C manuel de référence*, S.P. Harbison & G.L Steele JR
- *Le langage C ANSI*, B.W Kernighan & D.M. Richie
- *Borland Pascal 7*, Scott D. Palmer

- PLOTMTV
- Home page de GNUPLOT : http://www.gnuplot.org/
- Home page de DOXYGEN : http://www.stack.nl/~dimitri/doxygen/
- A2PS
- Home page de Aerologic, concepteur de CMARC : http://www.aerologic.com/index.html



Los picos del Infierno e los dos ibones inferiores de Pezico